IN THE CLAIMS

Claims 1-6 and 8-29 are pending.

Claim 7 is canceled.

Claims 1, 4, 10, 20, 29 are currently amended.

1. (Currently amended) A method of supporting ~~Supporting~~ and dynamically managing media pipeline topology changes during a media application session ~~sessions~~ to facilitate seamless presentation of media during dynamic changes, the method comprising:

receiving a partial media pipeline topology that defines how data flows through a plurality of nodes in the partial media pipeline topology including at least a first media source node and at least a first media sink node;

retrieving a cached media pipeline topology when the partial media pipeline topology is not sufficient to permit presentation to further define how data flows through a plurality of nodes in the partial media pipeline topology including at least a second media source node, at least a second media sink node, and at least one transform node;

[[and]]

cloning ~~copying~~ one or more nodes including state information from the cached media pipeline topology to the partial media pipeline topology during the media application session thus creating a full media pipeline topology to facilitate seamless presentation of media;

maintaining a data table that correlates one or more decoders or encoders in the cached media pipeline topology with one or more source nodes or destination nodes in the cached media pipeline topology;

associating a source node with a same instance of the decoder and requiring that a same decoder be used if a media source node is re-used in a subsequent topology or a destination node with the same instance of the encoder and requiring that the same encoder be used if a media destination node is re-used in a subsequent topology; and

facilitating the seamless presentation when receiving data from the media source node by starting a presentation clock after receiving data at the media sink node.

2. (Previously Presented) The method of claim 1, wherein the partial media pipeline topology is received from a remote process as a parameter in an interface call.

3. (Previously presented) The method of claim 1, wherein the cached media pipeline topology is retrieved as a parameter in an interface call.

4. (Currently amended) The method of claim 1, further comprising determining whether there are corresponding nodes in the partial media pipeline topology and the cached media pipeline topology.

5. (Previously Presented) The method of claim 4, further comprising transferring the at least one transform node from the cached media pipeline topology to the partial media pipeline topology.

6. (Previously Presented) The method of claim 1, further comprising cloning a plurality of connected nodes from the cached media pipeline topology into the partial media pipeline topology.

7. (Canceled).

8. (Previously Presented) The method of claim 1, further comprising connecting one or more nodes in the partial media pipeline topology.

9. (Previously Presented) The method of claim 8, wherein connecting the one or more nodes in the pipeline topology between the first media source node and the first media sink node comprises generating a data path between an output of a node and an input of an intermediate node.

10. (Currently amended) A system comprising:

one or more computer-readable storage media; and

a media engine embodied on the one or more computer-readable storage media and configured to communicatively interact with an application to seamlessly present a media presentation;

the media engine being configured to use:

a media session to generate a partial topology, the partial topology including one or more media sources, individual ones of which serve as a source

of media content, and one or more media sinks configured to sink a media stream; and

a topology loader to resolve the partial topology into a full media topology, wherein the topology loader is configured to clone ~~copy~~ one or more nodes including state information from a cached media topology to resolve the full media topology, and the topologies define a flow of data through the nodes and a presentation clock is started after receiving data at a node.

11. (Original) The system of claim 10, wherein the media session passes the partial topology to the topology loader as a parameter in an interface call.

12. (Previously Presented) The system of claim 10, wherein the media session passes the cached media topology to the topology loader as a parameter in an interface call.

13. (Previously Presented) The system of claim 10, wherein the topology loader is configured to determine whether there are corresponding nodes in the partial topology and the cached media topology.

14. (Original) The system of claim 10, wherein the topology loader is configured to clone one or more intermediate nodes from the cached media topology, and to connect the one or more intermediate nodes in a communication path between a media source and a media sink in a partial topology.

15. (Previously Presented) The system of claim 14, wherein the one or more intermediate nodes comprise a decoder for decoding an output of a source node.

16. (Original) The system of claim 14, wherein the one or more intermediate nodes comprises an encoder for encoding an input of a source node.

17. (Original) The system of claim 10, wherein the topology loader is configured to maintain a data table that associates one or more decoder nodes with a source node from one or more previous topologies.

18. (Original) The system of claim 10, wherein the topology loader maintains a data table that stores one or more encoder nodes from one or more previous topologies.

19. (Original) The system of claim 10, wherein the topology loader returns a fully resolved topology to the media session.

20. (Currently amended) One or more computer-readable storage media storing ~~comprising~~ computer executable instructions comprising instructions that, when executed on a computer, direct the computer to:

receive a partial media topology defined by the flow of data through various components that includes a plurality of nodes including at least a first media source node and at least a first media sink node;

retrieve a cached media topology that includes a plurality of nodes including at least a second media source node, at least a second media sink node, and at least one transform node; [[and]]

clone ~~copy~~ one or more nodes including state information from the cached media topology to a fully resolved media topology; and

start a presentation clock after receiving data at the media sink node.

21. (Previously Presented) The one or more computer-readable storage media of claim 20, wherein the partial media topology is received from a remote process as a parameter in an interface call.

22. (Previously Presented) The one or more computer-readable storage media of claim 20, wherein the cached media topology is retrieved as a parameter in an interface call.

23. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to determine whether there are corresponding nodes in the partial media topology and the cached media topology.

24. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a

computer, direct the computer to transfer the at least one transform node from the cached media topology to the partial media topology.

25. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to clone a plurality of connected nodes from the cached media topology into the partial media topology.

26. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to maintain a data table that correlates one or more decoders in the cached media topology with one or more source nodes in the cached media topology.

27. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to connect one or more nodes in the partial media topology.

28. (Previously Presented) The one or more computer-readable storage media of claim 20, further comprising computer executable instructions that, when executed on a computer, direct the computer to generate a data path between an output of an upstream node and an input of a downstream node.

29. (Currently amended) A topology loader module comprising computer executable instructions stored in computer-readable storage media that, when executed by a computer, provide:

means for receiving a partial media topology that defines how data flows through a plurality of nodes including at least a first media source node and at least a first media sink node;

means for retrieving a cached media topology that defines how data flows through a plurality of nodes including at least a second media source node, at least a second media sink node, and at least one transform node; [[and]]

means for associating the nodes with a same instance of their encoder or decoder and requiring a same encoder or decoder be re-used in a subsequent topology; and

means for cloning ~~copying~~ one or more nodes including state information from the cached media topology to a fully resolved media topology.